Presentation

RENTAL AGREEMENT SMART CONTRACT

Yohan Hwang
Darius Griffin
Bek Davronov

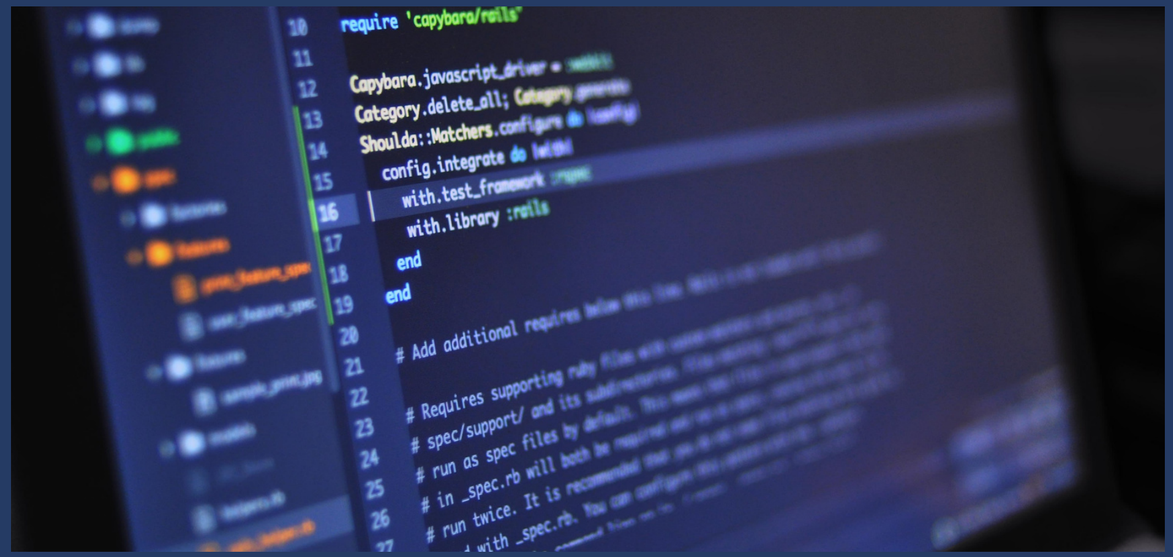
PROJECT OVERVIEW

- Create a smart contract for property managers that takes advantage of the capabilities of blockchain technology
- Landlords implement the smart contract with the key requirements of the tenancy such as rent amount, payment frequency, property and insurance details
- Once all the terms and conditions are reviewed, the contract is electronically signed by both parties and published on the Blockchain
- Once published, it is activated and executes transactions using a payments bridge to control the flow of funds between accounts, as specified in the terms of the contract.

PACKAGE REQUIREMENTS AND IMPORTS

We imported the following contract because it had basic rental agreement functions: *"https://github.com/kohshiba/ERC-X/blob/master/contracts/ERCX/Contract/ERCX.sol"*

We added our own requirements and functionalities on top of it to make it more efficient. The import was a bare bones simple rental lease agreement in which a tenant is approved, a token representing an apartment is transacted, and the balance of both landlord and tenant can be checked. We added more features which we believe are common in real life rental lease agreements, such as duration of lease, and approving a tenant based on a number of specific criteria as specified above.



Pinata imports:

- pinFiletoIPFS
- pinSONtoIPFS
- convertDatatoJSON

USER INSTRUCTIONS

- Download all the files from the repository
- Once download is complete, open the folder on VSCode
- Make sure to install Ethereum Remix plugin on your extensions
- Click on the Ethereum icon on the left and deploy the contract on the .sol file.
- Look to the right and copy and paste the deployed address and put it on the .env file for

“SMART_CONTRACT_DEPLOYED_ADDRESS” = “” between the quotations.

- On the .env file also the WEB_PROVIDER_URI = should be either HTTP://127.0.0.1:8545 or HTTP://127.0.0.1:7545
- Copy and paste the ABI from the right to the abi.json file
- Open a New Git Bash terminal.
- On the terminal, type Streamlit run app.py

RESULTS/EXAMPLES

Once up and running, the initial address is the landlord and only the initial address can execute the contract. Make sure it is set to that for any transactions. The first thing to do is to mint an apartment, which is a token representing a specific apartment.
After choosing lease duration and checking all the boxes, approve the tenant and then lease the apartment to another address. You can then check who owns it with some of the other buttons.


DEMO TIME

LIMITATIONS AND FUTURE DEVELOPMENTS

If we had more time, we could potentially connect the token to a smart lock device connected to the internet that could lock or unlock the apartment based on the contract regulations. If for instance someone leases the apartment, the entrance code to that lock would be given to the user by the contract. Once the contract expires, the lock combination would change.

This potentially could become a business model so that this web site becomes a platform where apartments are advertised by landlords and potential tenants browse apartments. And a lease could take place on the site itself almost automatically, much like Airbnb.

REFERENCES

- https://docs.soliditylang.org/en/v0.8.17/solidity-by-example.html
- ERC-X/ERCX.sol at master · kohshiba/ERC-X (github.com)
- Solidity by Example — Solidity 0.8.17 documentation (soliditylang.org)
- RollaProject/solidity-datetime: Gas-Efficient Solidity DateTime Library (github.com)

TEAM MEMBERS

Darius Griffin
Yohan Hwang
Bek Davronov

THANK YOU